KEY_Lesson10_Loops1

December 10, 2021

1 Loops

Earlier, we learned about lists, and how they are super useful ways to store a lot of information in one variable. In this lesson we will learn about the for loop, a powerful tool to perform commands on items in a list.

Suppose you had a list of animals and you wanted to print your list so that each item was on its own line.

[1]:

```
# create a list of animals
animals = ['turtle', 'penguin', 'platypus', 'zebra', 'giraffe']
```

[2]:

```
# print animals
print(animals)
```

```
['turtle', 'penguin', 'platypus', 'zebra', 'giraffe']
```

We see that using the `print` function on our list prints all the values in brackets and separated by commas on the same line. This isn't quite what we want. But, Python has a useful tool we can use for this problem called the for loop.

A for loop has three key parts:

```
for [item] in [list of items]:
  [command]
```

A for loop will look at each individual item in the list it is given, then run the given command on each item.

Let's take a look at how we might use a for loop to solve our problem

[3]:

```
# create a for loop to print each item in our animals list
for animal in animals:
    print(animal)
```

```
turtle
penguin
platypus
zebra
giraffe
```

A Python function that is really useful when writing loops is the `range` function. `range` creates a list of numbers between the two numbers given to the function.

If we use the output of the `range` function in a for loop, then whatever code we write in our for loop will execute on each number between the two parameters of `range`.

[4]:

```python
# use a for loop to print each number in range(0,5)
for value in range(0,5):
    print(value)
```

```
0
1
2
3
4
```

The `range` function can be useful when working with lists. To iterate over a whole list, we can use the `range` function with 0 as the start and the list length as the end. Let's test this out with our animals list!

[5]:

```python
# use a for loop and the range function to print every animal in the list animals
for index in range(0,len(animals)):
    print(animals[index])
```

```
turtle
penguin
platypus
zebra
giraffe
```

We don't just have to use range, though. Let's say we only wanted to print out some of the animals in our list. We could give our for loop a list of indices in our list we wanted to print

[6]:

```python
# create a list called indices containing indices of the list animals to print
indices = [0,1,4]

# use a for loop to only print animals at the above indices
for index in indices:
    print(animals[index])
```

```
turtle
penguin
giraffe
```

We can also put loops inside of each other. These are called **nested for loops**. Let's see what this might look like:

```
for [item1] in [list1]:
  for [item2] in [list2]:
    [command using item1 and item2]
```

It is very important to remember that item1 and item2 have different names. Otherwise, Python might not know which list you want to iterate through, and your code will not run quite how you expected!

Nested for loops are useful in a lot of contexts, but they are especially useful when we have a list inside of a list. This is often referred to as a `2D list`.

[7]:

```
# create a list of names for our animals (remember, we have 5 animals!)
names = ['Shelly', 'Tuxedo', 'Perry', 'Stripes', 'Bob']

# create a 2D list combining animals and names
zoo = [animals, names]

# print zoo so we can look at its structure
print(zoo)
```

```
[['turtle', 'penguin', 'platypus', 'zebra', 'giraffe'], ['Shelly', 'Tuxedo',
'Perry', 'Stripes', 'Bob']]
```

Now we have a list of lists! What do you think it would look like if we pulled out the first element of `zoo`?

[8]:

```
# command python to get the first element in the list called zoo
zoo[0]
```

[8]:

```
['turtle', 'penguin', 'platypus', 'zebra', 'giraffe']
```

We can see that using index zero to subset `zoo` gets the first list. What if we wanted to get the second element from `zoo[0]`?

[9]:

```
# command python to get the second element of the first element of zoo
zoo[0][1]
```

[9]:

```
'penguin'
```

Say we want to print our whole list so that each animal has its name printed next to it. Something like this:

```
turtle Shelly
penguin Tuxedo
platypus Perry
zebra Stripes
giraffe Bob
```

This can be done using nested for loops!

[10]:

```
# use a nested for loop to print each animal and its name
for row in range(0,5):

    species_and_name = ""

    for column in range(0,2):
        species_and_name = species_and_name + zoo[column][row] + " "

    print(species_and_name)
```

```
turtle Shelly
penguin Tuxedo
platypus Perry
zebra Stripes
giraffe Bob
```

Excellent work! You just learned about for loops. You learned: - How to write a for loop to iterate over all items in a list - How to use the `range` function - How to create a list of lists, or **2D list** - How to write nested for loops